

```
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGGGG  RRRRRRRRRRRR  TTTTTTTTTTTTTT  LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMMMMM  MMMMMM  GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSS            MMM      MMM      GGG            RRR      RRR      TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSSSSSSSSSS    MMM      MMM      GGG            RRRRRRRRRRRR  TTT      LLL
SSS            SSS      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGGGGGGGGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSS            SSS      MMM      GGG      GGG      GGG  RRR      RRR      TTT      LLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG      RRR      RRR      TTT      LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG      RRR      RRR      TTT      LLLLLLLLLLLLLLLL
SSSSSSSSSSSSS  MMM      MMM      GGGGGGGGGG      RRR      RRR      TTT      LLLLLLLLLLLLLLLL
```



```

SSSSSSSSS MM MM GGGGGGGG TTTTTTTTTT P P P P P P P A A A A A A LL I I I I I 8 8 8 8 8 8 8
SSSSSSSSS MM MM GGGGGGGG TTTTTTTTTT P P P P P P P A A A A A A LL I I I I I 8 8 8 8 8 8 8
SS SS MMMM MMMM GG TT PP PP PP AA AA LL I I 8 8 8 8 8 8 8
SS SS MMMM MMMM GG TT PP PP PP AA AA LL I I 8 8 8 8 8 8 8
SS SS MM MM GG TT PP PP PP AA AA LL I I 8 8 8 8 8 8 8
SSSSSSS MM MM GG TT P P P P P P P AA AA LL I I 8 8 8 8 8 8 8
SSSSSSS MM MM GG TT P P P P P P P AA AA LL I I 8 8 8 8 8 8 8
SS SS MM MM GG GGGGGG TT PP GGGGGG AA AAAAAAAAAA LL I I 8 8 8 8 8 8 8
SS SS MM MM GG GGGGGG TT PP GGGGGG AA AAAAAAAAAA LL I I 8 8 8 8 8 8 8
SS SS MM MM GG GG GG TT PP AA AA LL I I 8 8 8 8 8 8 8
SSSSSSSSS MM MM GGGGGG TT PP AA AA LL I I 8 8 8 8 8 8 8
SSSSSSSSS MM MM GGGGGG TT PP AA AA LLLLLLLLLL I I I I I 8 8 8 8 8 8 8
SSSSSSSSS MM MM GGGGGG TT PP AA AA LLLLLLLLLL I I I I I 8 8 8 8 8 8 8

```


%TITLE 'SMGTPALIB - SMG\$ TPARSE require files'
Run-Time Library Screen Management Bliss Definition Library
Module: SMGTPALIB.REQ Edit: PLL1001

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

++
FACILITY: Screen Management

ABSTRACT:

This file contains require files used in the creation of TERMTABLE.EXE.

MODIFIED BY:

1-001 - Original. PLL 30-JAN-1984

LIBRARY 'RTLSTARLE';

! Starlet definitions

REQUIRE 'RTLIN:SMGTABDEF';

! Terminal Table Definitions

Terminal Table Data Structure Definitions for RTL SMG\$ facility
File: SMGTABDEF.REQ Edit: PLL1001

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

++
FACILITY: Screen Management

ABSTRACT:

This module defines the internal structure of the binary
TERMTABLE.EXE.

MODIFIED BY:

1-001 - Original. PLL 2-Nov-1983

The internal structure of TERMTABLE.EXE is divided into several sections.

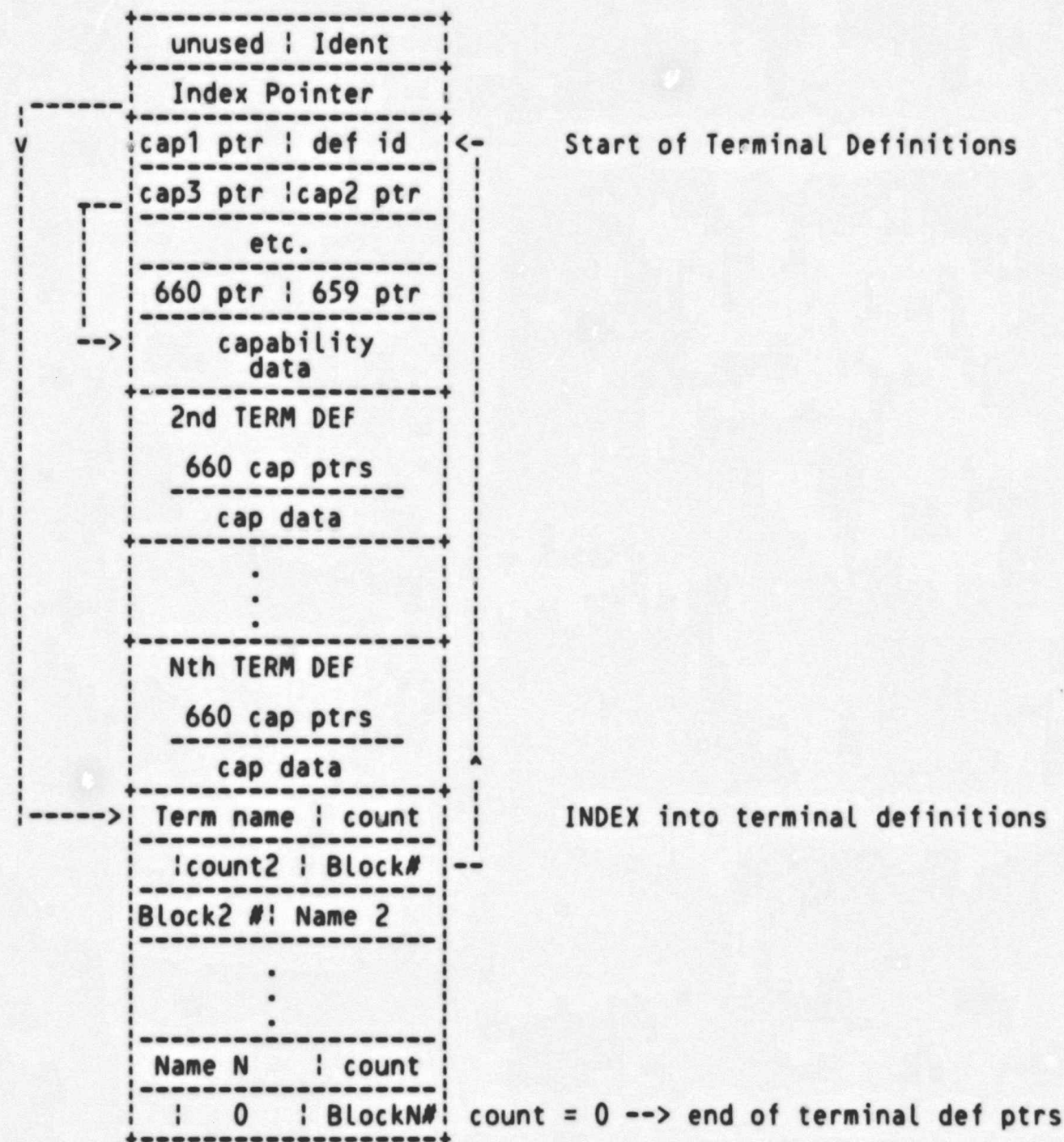
Each terminal definition consists of 660 capability pointers and
a data area. If a terminal does not have a particular capability defined,
the pointer for that capability will be zero. Otherwise, the capability
pointer is an offset into the data area.

Each capability data consists of a count and the data. The data may be
a binary number or an ascii string. A negative count indicates that
this is a dynamic string which requires processing (substitution, conversion,
or arithmetic).

TERMTABLE.EXE also contains an index so that individual terminal definitions

can be located. The index will consist of a count, the ascii name for the terminal, and the block number where the terminal definition begins. The index will actually be located at the end of TERMTABLE.EXE. This is because the number of terminal definitions is unpredictable, and we don't want to impose an arbitrary size constraint that would restrict the number of terminals defined.

TERMTABLE.EXE will begin with an ident number, to allow for future changes. These future changes would probably allow more terminal capabilities (greater than 660).




```

R0158 0 !
R0159 0
R0160 0
R0161 0 !+
R0162 0 !- Offsets used to access a terminal table.
R0163 0
R0164 0 MACRO
R0165 0     TTB_W_IDENT          = 0, 0, 16, 0%, ! ident field (to allow for
R0166 0     TTB_W_unused        = 2, 0, 16, 0%, ! future changes)
R0167 0     TTB_L_INDEX_OFFSET = 4, 0, 32, 0%, ! not used
R0168 0
R0169 0     TTB_L_INDEX_OFFSET = 4, 0, 32, 0%, ! offset from here to terminal
R0170 0     TTB_L_INDEX_OFFSET = 4, 0, 32, 0%, ! index
R0171 0
R0172 0 !+
R0173 0 !- Define constants needed to build terminal table
R0174 0
R0175 0 LITERAL
R0176 0     SMG$K_HEADER_SIZE      = 512,          ! 1 block for header info
R0177 0     SMG$K_TERM_INDEX_SIZE = 5000,         ! arbitrary size for index
R0178 0     SMG$K_CAP_PTRS_SIZE   = 1536,         ! 3 blocks to hold ptrs
R0179 0     SMG$K_CAP_DATA_SIZE   = 5120,         ! 10 blocks to hold data
R0180 0     SMG$K_TERM_DEF_SIZE   = 6656,         ! ptrs + data (3+10 blocks)
R0181 0     SMG$K_TERM_DEF_ID     = 137,          ! random number to identify
R0182 0     SMG$K_TERM_DEF_ID     = 137,          ! a valid terminal def
R0183 0     SMG$K_TERM_DEF_ID_OFFSET = 0,          ! location of id - 1st
R0184 0     SMG$K_TERM_DEF_ID_OFFSET = 0,          ! word in ptr vector
R0185 0     SMG$K_MIN_REQUEST_CODE = 1,            ! lowest capability number
R0186 0     SMG$K_MIN_REQUEST_CODE = 1,            !
R0187 0     SMG$K_MAX_REQUEST_CODE = 660,         ! highest capability number
R0188 0
R0189 0 !+
R0190 0 !- The following are used when parsing arithmetic expressions.
R0191 0 !- Information about expressions is stored via these encodings.
R0192 0
R0193 0 LITERAL
R0194 0     SMG$K_FAO_STRING        = %X'FFFFFFFF',
R0195 0     SMG$K_ARITH_STRING     = %X'FFFFFFFFE',
R0196 0     SMG$K_OPERAND          = %X'FFFFFFFFD',
R0197 0     SMG$K_SUBSTITUTE      = %X'FFFFFFFFC',
R0198 0     SMG$K_ADD              = %X'FFFFFFFFB',
R0199 0     SMG$K_SUBTRACT        = %X'FFFFFFFFA',
R0200 0     SMG$K_MULTIPLY       = %X'FFFFFFFF9',
R0201 0     SMG$K_DIVIDE          = %X'FFFFFFFF8',
R0202 0     SMG$K_STORE            = %X'FFFFFFFF7';

```


SMGTPALIB - SMGS TPARSE require files

G 16
15-Sep-1984 23:28:41
15-Sep-1984 22:51:35

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGTPALIB.REQ;1

Page 5
(1)

: 0203 0
: 0204 0

REQUIRE 'RTLIN:SMGTPACTL';

! TPARSE parameter block defs

! SMGTPACTL.REQ - SMG TPARSE Control Block Defs
! Version 1-002 - Edit: PLL1001

```
*****
*
*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
*  ALL RIGHTS RESERVED.
*
*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
*  TRANSFERRED.
*
*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
*  CORPORATION.
*
*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*
*****
```

! AUTHOR: P. Levesque

! EDIT HISTORY:
! 1-001 - Original. PLL 28-Nov-1983

LITERAL

SMG\$K_PARAM_BLOCK_SIZE = TPA\$K_LENGTH0 + 60; ! TPARSE parameter block length

+
Two parameter blocks are actually allocated together and are adjacent.
The first is the TPARSE parameter block, and the second is a private
parameter block needed to pass info between the routines in SMGBLDTRM.B32.
The TPARSE action routines in SMGTPATAB.B32 also need these variables.

TPARSE symbols (TPA\$ xxx) are defined in RTLTPAMAC.REQ. Only private
symbols are defined here.

MACRO

PARAM_A_TXT_FAB	= TPA\$K_LENGTH0 + 0, 0, 32, 0%
	! addr of FAB for TERMABLE.TXT
PARAM_A_TXT_RAB	= TPA\$K_LENGTH0 + 4, 0, 32, 0%
	! addr of RAB for TERMABLE.TXT
PARAM_A_BINARY_FAB	= TPA\$K_LENGTH0 + 8, 0, 32, 0%
	! addr of FAB for TERMABLE.EXE
PARAM_A_BINARY_RAB	= TPA\$K_LENGTH0 + 12, 0, 32, 0%
	! addr of RAB for TERMABLE.EXE


```

: R0262 0      PARAM_A_CAP_PTRS      = TPASK_LENGTH0 +16, 0, 32, 0%,
: R0263 0      |      |      |      |      |      |      |      |      |      |
: R0264 0      PARAM_A_CAP_DATA      = TPASK_LENGTH0 +20, 0, 32, 0%,
: R0265 0      |      |      |      |      |      |      |      |      |      |
: R0266 0      PARAM_A_HEADER        = TPASK_LENGTH0 +24, 0, 32, 0%,
: R0267 0      |      |      |      |      |      |      |      |      |      |
: R0268 0      PARAM_L_TERM_INDEX_SIZE = TPASK_LENGTH0 +28, 0, 32, 0%,
: R0269 0      |      |      |      |      |      |      |      |      |      |
: R0270 0      PARAM_A_TERM_INDEX    = TPASK_LENGTH0 +32, 0, 32, 0%,
: R0271 0      |      |      |      |      |      |      |      |      |      |
: R0272 0      |      |      |      |      |      |      |      |      |      |
: R0273 0      PARAM_L_CUR_TERM_DEF   = TPASK_LENGTH0 +36, 0, 32, 0%,
: R0274 0      |      |      |      |      |      |      |      |      |      |
: R0275 0      PARAM_L_CUR_DATA_BYTE  = TPASK_LENGTH0 +40, 0, 32, 0%,
: R0276 0      |      |      |      |      |      |      |      |      |      |
: R0277 0      PARAM_L_CUR_CAP_NUMBER = TPASK_LENGTH0 +44, 0, 32, 0%,
: R0278 0      |      |      |      |      |      |      |      |      |      |
: R0279 0      |      |      |      |      |      |      |      |      |      |
: R0280 0      PARAM_L_SAVED_TOKENCNT = TPASK_LENGTH0 +48, 0, 32, 0%,
: R0281 0      |      |      |      |      |      |      |      |      |      |
: R0282 0      |      |      |      |      |      |      |      |      |      |
: R0283 0      PARAM_L_SAVED_TOKENSTR = TPASK_LENGTH0 +52, 0, 32, 0%,
: R0284 0      |      |      |      |      |      |      |      |      |      |
: R0285 0      |      |      |      |      |      |      |      |      |      |
: R0286 0      PARAM_L_ORIG_TXT       = TPASK_LENGTH0 +56, 0, 32, 0%;
: R0287 0      |      |      |      |      |      |      |      |      |      |
: R0288 0      |      |      |      |      |      |      |      |      |      |
: R0289 0      ! End of SMGTPACTL.REQ |      |      |      |      |      |      |

```


SMGTPALIB - SMGS TPARSE require files

J 16
15-Sep-1984 23:28:41
15-Sep-1984 22:51:35

VAX-11 Bliss-32 V4.0-742
[SMGRTL.SRC]SMGTPALIB.REQ;1

Page 8
(1)

: 0290 0
: 0291 0

REQUIRE 'RTLIN:SMGTRMMAC';

! TPARSE macros


```
R0292 0 XTITLE 'SMGTRMMAC.REQ - Macros for TERMTABLE'
R0293 0 Module: SMGTRMMAC.REQ Edit: PLL1001
R0294 0
R0295 0 *****
R0296 0 *
R0297 0 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
R0298 0 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
R0299 0 * ALL RIGHTS RESERVED.
R0300 0 *
R0301 0 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
R0302 0 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
R0303 0 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
R0304 0 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
R0305 0 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
R0306 0 * TRANSFERRED.
R0307 0 *
R0308 0 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
R0309 0 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
R0310 0 * CORPORATION.
R0311 0 *
R0312 0 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
R0313 0 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
R0314 0 *
R0315 0 *
R0316 0 *****
R0317 0
R0318 0
R0319 0 ++
R0320 0 FACILITY: Screen Management
R0321 0
R0322 0 ABSTRACT:
R0323 0
R0324 0 This require file contains macros used in the creation of TERMTABLE.EXE.
R0325 0
R0326 0 MODIFIED BY:
R0327 0
R0328 0 1-001 - Original. PLL 30-JAN-1984
R0329 0 --
R0330 0
R0331 0 +
R0332 0 Macro $INCR_CAP_STRING_SIZE
R0333 0
R0334 0 Add the specified number of bytes to the capability string size.
R0335 0 We may be dealing with a negative size if extra processing is needed.
R0336 0 -
R0337 0
R0338 0 MACRO $INCR_CAP_STRING_SIZE (NUMBER_BYTES, STRING_SIZE) =
R0339 0 BEGIN
R0340 0 IF .STRING_SIZE LSS 0
R0341 0 THEN
R0342 0 STRING_SIZE = .STRING_SIZE - NUMBER_BYTES
R0343 0 ELSE
R0344 0 STRING_SIZE = .STRING_SIZE + NUMBER_BYTES;
R0345 0 END;%
R0346 0
R0347 0 !+
R0348 0 ! Macro $INCR_CUR_DATA_BYTE
```



```

R0349 0
R0350 0
R0351 0
R0352 0
R0353 0
R0354 0
R0355 0
R0356 0
R0357 0
R0358 0
R0359 0
R0360 0
R0361 0
R0362 0
R0363 0
R0364 0
R0365 0
R0366 0
R0367 0
R0368 0
R0369 0
R0370 0
R0371 0
R0372 0
R0373 0
R0374 0
R0375 0
R0376 0
R0377 0
R0378 0
R0379 0
R0380 0
R0381 0
R0382 0
R0383 0
R0384 0
R0385 0
R0386 0
R0387 0
R0388 0
R0389 0
R0390 0
R0391 0
R0392 0
R0393 0
R0394 0
R0395 0
R0396 0
R0397 0
R0398 0
R0399 0
R0400 0
R0401 0
R0402 0
R0403 0
R0404 0
R0405 0

|
| Update the currently available data byte by the number of bytes
| just written. If this is the first byte of the string, then we
| also wrote a byte count.
|
| -
|
MACRO $INCR_CUR_DATA_BYTE (NUMBER_BYTES, FIRST_FLAG) =
BEGIN
  AP [PARAM_L_CUR_DATA_BYTE] = .AP [PARAM_L_CUR_DATA_BYTE] +
    NUMBER_BYTES +
    (IF FIRST_FLAG
     THEN 2      ! 1 for size, 1 for kind
     ELSE 0);

  SMG$$DATA_OFFSET = .SMG$$DATA_OFFSET + NUMBER_BYTES +
    (IF FIRST_FLAG
     THEN 2
     ELSE 0);

END; %;

| +
| Macro to update the size of an arithmetic capability. This size
| will be negative.
|
MACRO $UPDATE_ARITH_STRING_SIZE (NUMBER_BYTES) =
BEGIN
  BIND
  CAP_PTRS = .AP [PARAM_L_CUR_TERM_DEF] : VECTOR [,WORD];
  LOCAL
  CAP_SIZE : REF VECTOR [,BYTE,SIGNED];

  CAP_SIZE = .AP [PARAM_L_CUR_TERM_DEF] +
    .CAP_PTRS [.AP [PARAM_L_CUR_CAP_NUMBER]];

  CAP_SIZE [0] = .CAP_SIZE [0] - NUMBER_BYTES;
END; %;

| +
| We fill buffers with data and then do a block I/O write when we reach
| the end of a definition or when the buffer is full. Here we check to
| see if the buffer is full (ie. if there is room for the number of bytes
| we are about to put in it).
|
| -
|
MACRO $CHECK_BUFFER_OVERFLOW (BYTES_TO_WRITE) =
BEGIN
  LOCAL
  REMAINING_BYTES;
  REMAINING_BYTES = SMG$K_CAP_DATA_SIZE -
    (.AP [PARAM_L_CUR_DATA_BYTE] - .AP [PARAM_A_CAP_DATA]);
  IF BYTES_TO_WRITE GTR .REMAINING_BYTES
  THEN
    CALLG (.AP, SMG$$WRITE_DATA);
END; %;

```


: R0406 0
: R0407 0 ! End of file SMGTRMMAC.REQ


```
: 0408 0
: 0409 0 ! End of file SMGTPALIB.REQ
```

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	1	0	581	00:01.0

COMMAND QUALIFIERS

```
: BLISS/LIBRARY=LIB$:SMGTPALIB/LIST=LISS$:SMGTPALIB/SOURCE=REQUIRE SRC$:SMGTPALIB
: Run Time: 00:04.0
: Elapsed Time: 00:05.7
: Lines/CPU Min: 6135
: Lexemes/CPU-Min: 9780
: Memory Used: 32 pages
: Library Precompilation Complete
```


0361

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY